

Fondamenti di Informatica T-1, 2019/2020 – Modulo 2

Prova d'Esame 5A di Giovedì 9 Luglio 2020 – tempo a disposizione 2h

Avvertenze per la consegna: apporre all'inizio di ogni file sorgente un commento contenente i propri dati (cognome, nome, numero di matricola) e il **numero** della prova d'esame. Al termine, **consegnare tutti i file sorgenti** necessari alla compilazione e alla corretta esecuzione del programma.

Nota: il **main** non è opzionale; i **test** richiesti vanno implementati.

Consiglio: per verificare l'assenza di *warning*, eseguire sempre *"Rebuild All"*.

Valerio gestisce con orgoglio il suo bar Rhythm&Blues, e ha deciso recentemente di ampliare i suoi servizi offrendo panini farciti realizzati su misura dietro richieste dei clienti. Per prima cosa ha informatizzato la lista degli ingredienti disponibili, scrivendo su un file di testo le informazioni. In particolare, memorizza un ingrediente su ogni riga del file di testo (ma è un po' sbadato, ogni tanto lascia delle righe vuote). Per ogni ingrediente, memorizza: il **nome** dell'ingrediente (una stringa di al più 63 caratteri utili, senza spazi); a seguire, separata da uno spazio, la **quantità** di quell'ingrediente, espressa in grammi (un intero); infine, ancora separato da uno spazio, il **costo** in Eur/kg di quell'ingrediente (un float).

Per limitare il tempo speso a fare i panini, Il barista Valerio ha deciso che ogni panino potrà contenere al massimo 3 ingredienti (ma potrebbe contenerne meno); inoltre, memorizza le richieste dei clienti scrivendo su un file di testo, su ogni riga, il nome del **cliente** (stringa di al più 63 caratteri utili, senza spazi), e a seguire i tre ingredienti, tutti separati da spazi (gli ingredienti sono indicati solo con il loro nome). Ogni lista degli ingredienti è sempre terminata dalla stringa "fine"

"ingredienti.txt": prosciutto 80 26.50 funghi_trifolati 20 35.00 mozzarella 50 9.00 scamorza 15 18.00 speck 80 25.99 prosciutto_cotto 80 19.90	"panini.txt": Federico prosciutto funghi_trifolati fine Piero scamorza speck fine Paola mozzarella prosciutto_cotto fine Federico speck rucola fine
---	--

Esercizio 1 – Strutture dati Ingrediente e Panino, e funzioni di lett./scritt. (mod. element.h e bar.h/c)

Si definisca una struttura dati **Ingrediente** per memorizzare i dati relativi a un ingrediente come sopra descritto (nome, quantità, e costo). Si definisca poi una struttura dati **Panino** per memorizzare i dati relativi ad un panino come sopra descritto (nome del cliente, array di strutture dati Ingrediente grandi al massimo tre, numero degli ingredienti effettivamente presenti).

Si definisca la funzione:

```
Ingrediente * leggiIng(char * fileName, int * dim);
```

che, ricevuto in ingresso il nome di un file contenente gli ingredienti, e un intero passato per riferimento, legga i dati contenuti nel file e li restituisca tramite un array di strutture dati di tipo **Ingrediente**, allocato dinamicamente e della dimensione minima necessaria (non è noto a priori quanti ingredienti siano memorizzati nel file). Tramite l'intero **dim** passato per riferimento la funzione dovrà restituire la dimensione dell'array. Qualora vi siano problemi nella lettura, la funzione restituisca un puntatore a NULL e dimensione pari a zero.

Si definisca poi una procedura ricorsiva:

```
void stampa(Ingrediente * elenco, int dim);
```

che, ricevuto in ingresso un puntatore a un array di strutture dati di tipo **Ingrediente**, e la dimensione di tale array, stampi a video l'elenco degli ingredienti, in maniera ricorsiva.

Si definisca la funzione:

Fondamenti di Informatica T-1, 2019/2020 – Modulo 2

Prova d'Esame 5A di Giovedì 9 Luglio 2020 – tempo a disposizione 2h

Ingrediente trova(Ingrediente * elenco, int dim, char * nome);

che, ricevuto in ingresso un array di strutture dati **Ingrediente** e la dimensione di tale array, e il nome di un ingrediente da cercare, restituisca la struttura dati corrispondente all'ingrediente specificato tramite il suo nome. Qualora l'ingrediente cercato non sia presente nell'array, la funzione dovrà restituire una struttura dati **Ingrediente** con nome pari alla stringa "NULL".

Il candidato abbia cura di realizzare nel main opportuni test al fine di verificare il corretto funzionamento delle funzioni di cui sopra, avendo cura di deallocare la memoria, se necessario.

Esercizio 2 – Ordinamento e lettura panini (moduli element.h/c e bar.h/c)

Si definisca la procedura:

void ordina(Ingrediente * v, int dim);

che, ricevuta in ingresso un array di strutture dati di tipo **Ingrediente**, e di dimensione **dim**, provveda ad ordinare l'array in base al seguente criterio: in ordine crescente in base alla quantità; a parità di quantità, in ordine decrescente in base al prezzo. Per l'ordinamento si utilizzi un algoritmo a scelta tra quelli visti a lezione.

Si definisca la funzione:

list leggiPanini(char * fileName, Ingrediente * elenco, int dim);

che, ricevuta in ingresso il nome di un file contenente i panini richiesti dai clienti, e l'elenco degli ingredienti (nella forma di un array di strutture dati **Ingrediente** e la sua dimensione), legga da file e restituisca una lista di strutture dati **Panino** contenente i dati nel file.

Si noti che il file delle richieste clienti contiene, per ogni panino, gli ingredienti nella forma del loro nome. Ma per formare correttamente la struttura dati **Panino**, è necessario avere a disposizione le strutture dati **Ingrediente** (e non solo il loro nome). A tal scopo, la funzione riceve in ingresso anche l'elenco degli ingredienti, da cui dato il nome di un ingrediente è possibile recuperare la corrispondente struttura dati (tramite la funzione **trova(...)** definita precedentemente).

In caso di problemi nella lettura da file, la funzione dovrà restituire una lista vuota.

Esercizio 3 – Eliminazione duplicati (moduli element.h/c e bar.h/bar.c)

Si definisca la funzione:

list eliminaDup(list panini);

che, ricevuta in ingresso la lista contenente i panini richiesti dagli utenti, restituisce una nuova lista in cui eventuali richieste duplicate siano state eliminate.

Una richiesta di panino è il duplicato di un'altra se sono relative allo stesso cliente e hanno lo stesso numero di ingredienti (che possono eventualmente essere però diversi). In caso di panini duplicati, nella lista di uscita si potrà inserire a scelta il primo panino che si incontra scorrendo la lista in ingresso, l'ultimo, o altro panino a scelta.

Esercizio 4 -- Stampa dei risultati, e de-allocazione memoria (main.c)

Il candidato realizzi nella funzione **main(...)** un programma che legga dai file i dati relativi agli ingredienti e ai panini. Dopo aver eliminato eventuali panini duplicati, il programma stampi a video quanti panini ci sono in lista, e poi ne stampi l'elenco con anche gli ingredienti.

Al termine del programma, il candidato abbia cura di de-allocazione tutta la memoria allocata dinamicamente, ivi compresa la memoria allocata per le liste, se possibile.

Fondamenti di Informatica T-1, 2019/2020 – Modulo 2

Prova d'Esame 5A di Giovedì 9 Luglio 2020 – tempo a disposizione 2h

“element.h”:

```
#pragma once
#define _CRT_SECURE_NO_WARNINGS

#ifndef _ELEMENT_H
#define _ELEMENT_H

#include <string.h>

#define DIM_NOME 64

typedef struct {
    char nome[DIM_NOME];
    int quant;
    float prezzo;
} Ingrediente;

typedef struct {
    char cliente[DIM_NOME];
    Ingrediente ing[3];
    int dim;
} Panino;
typedef Panino element;

int compare(Ingrediente e1, Ingrediente e2);

int equals(Panino p1, Panino p2);

#endif
```

“element.c”:

```
#include "element.h"

int compare(Ingrediente e1, Ingrediente e2) {
    int result;
    result = e1.quant - e2.quant;
    if (result == 0) {
        if (e1.prezzo > e2.prezzo)
            result = -1;
        else
            if (e1.prezzo < e2.prezzo)
                result = 1;
            else
                result = 0;
    }
    return result;
}

int equals(Panino p1, Panino p2) {
    return !strcmp(p1.cliente, p2.cliente) && p1.dim == p2.dim;
}
```

Fondamenti di Informatica T-1, 2019/2020 – Modulo 2

Prova d'Esame 5A di Giovedì 9 Luglio 2020 – tempo a disposizione 2h

"list.h":

```
#pragma once
#ifndef LIST_H
#define LIST_H

#include "element.h"

typedef struct list_element {
    element value;
    struct list_element *next;
} item;

typedef item* list;

typedef int boolean;

/* PRIMITIVE */
list emptylist(void);
boolean empty(list);
list cons(element, list);
element head(list);
list tail(list);

void showlist(list l);
void freelist(list l);
int member(element el, list l);

//list insord_p(element el, list l);

#endif
```

"list.c:"

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include "list.h"

/* OPERAZIONI PRIMITIVE */
list emptylist(void) /* costruttore lista vuota */
{
    return NULL;
}

boolean empty(list l) /* verifica se lista vuota */
{
    return (l == NULL);
}

list cons(element e, list l)
{
    list t; /* costruttore che aggiunge in testa alla lista */
    t = (list)malloc(sizeof(item));
    t->value = e;
    t->next = l;
    return(t);
}

element head(list l) /* selettore testa lista */
{
    if (empty(l)) exit(-2);
    else return (l->value);
}
```

Fondamenti di Informatica T-1, 2019/2020 – Modulo 2

Prova d'Esame 5A di Giovedì 9 Luglio 2020 – tempo a disposizione 2h

```
list tail(list l)          /* selettore coda lista */
{
    if (empty(l)) exit(-1);
    else return (l->next);
}

void showlist(list l) {
    element temp;
    int i;
    if (!empty(l)) {
        temp = head(l);
        printf("%s:\n",
            temp.cliente);
        for (i = 0; i < temp.dim; i++)
            printf("\t%s\n", temp.ing[i].nome);
        showlist(tail(l));
        return;
    }
    else {
        printf("\n\n");
        return;
    }
}

int member(element el, list l) {
    int result = 0;
    while (!empty(l) && !result) {
        result = equals(el, head(l));
        if (!result)
            l = tail(l);
    }
    return result;
}

void freelist(list l) {
    if (empty(l))
        return;
    else {
        freelist(tail(l));
        free(l);
    }
    return;
}
```

Fondamenti di Informatica T-1, 2019/2020 – Modulo 2

Prova d'Esame 5A di Giovedì 9 Luglio 2020 – tempo a disposizione 2h

"bar.h":

```
#pragma once

#define _CRT_SECURE_NO_WARNINGS

#ifndef _BAR_H
#define _BAR_H

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "element.h"
#include "list.h"

// Es. 1
Ingrediente * leggiIng(char * fileName, int * dim);
void stampa(Ingrediente * elenco, int dim);
Ingrediente trova(Ingrediente * elenco, int dim, char * nome);

// Es. 2
void ordina(Ingrediente * v, int dim);
list leggiPanini(char * fileName, Ingrediente * elenco, int dim);

// Es. 3
list eliminaDup(list panini);

#endif
```

"bar.c":

```
#include "bar.h"

// Es. 1
Ingrediente * leggiIng(char * fileName, int * dim) {
    FILE * fp;
    Ingrediente * result;
    Ingrediente temp;
    int counter;

    *dim = 0;
    result = NULL;

    fp = fopen(fileName, "rt");
    if (fp == NULL) {
        printf("Errore nell'apertura di %s\n", fileName);
        return result;
    }
    else {
        counter = 0;
        while (fscanf(fp, "%s%d%f", temp.nome, &(temp.quant), &(temp.prezzo)) == 3)
            counter++;

        if (counter > 0) {
            rewind(fp);
            result = (Ingrediente *) malloc(sizeof(Ingrediente) * counter);
            while (fscanf(fp, "%s%d%f", temp.nome, &(temp.quant), &(temp.prezzo))
== 3) {
                result[*dim] = temp;
                *dim = *dim + 1;
            }
        }
        fclose(fp);
        return result;
    }
}
```

Fondamenti di Informatica T-1, 2019/2020 – Modulo 2

Prova d'Esame 5A di Giovedì 9 Luglio 2020 – tempo a disposizione 2h

```
}

void stampa(Ingrediente * elenco, int dim) {
    if (dim <= 0)
        return;
    else {
        stampa(elenco, dim - 1);
        printf("%s:\t\t%dgr.\t\t%feur/kg.\n",  elenco[dim - 1].nome,  elenco[dim -
1].quant, elenco[dim - 1].prezzo);
    }
}

Ingrediente trova(Ingrediente * elenco, int dim, char * nome) {
    Ingrediente result;
    int i;
    int trovato = 0;

    for (i = 0; i < dim && !trovato; i++) {
        if (strcmp(elenco[i].nome, nome) == 0) {
            trovato = 1;
            result = elenco[i];
        }
    }
    if (!trovato)
        strcpy(result.nome, "NULL");
    return result;
}

// Es. 2
void scambia(Ingrediente * e1, Ingrediente * e2) {
    Ingrediente tmp = *e1;
    *e1 = *e2;
    *e2 = tmp;
}

void bubbleSort(Ingrediente v[], int n) {
    int i, ordinato = 0;
    while (n > 1 && !ordinato) {
        ordinato = 1;
        for (i = 0; i < n - 1; i++)
            if (compare(v[i], v[i + 1]) > 0) {
                scambia(&v[i], &v[i + 1]);
                ordinato = 0;
            }
        n--;
    }
}

void ordina(Ingrediente * v, int dim) {
    bubbleSort(v, dim);
}

list leggiPanini(char * fileName, Ingrediente * elenco, int dim) {
    list result;
    FILE * fp;
    Panino temp;
    char nome[DIM_NOME];

    result = emptylist();
    fp = fopen(fileName, "rt");
    if (fp == NULL)
        return result;
    else {
        while (fscanf(fp, "%s", temp.cliente) == 1) {
            temp.dim = 0;
```

Fondamenti di Informatica T-1, 2019/2020 – Modulo 2

Prova d'Esame 5A di Giovedì 9 Luglio 2020 – tempo a disposizione 2h

```
3) {
    while (fscanf(fp, "%s", nome)==1 && strcmp(nome, "fine") && temp.dim <
        temp.ing[temp.dim] = trova(elenco, dim, nome);
        temp.dim++;
    }
    result = cons(temp, result);
}
fclose(fp);
}
return result;
}

// Es. 3
list eliminaDup(list panini) {
    list result;
    Panino temp;

    result = emptylist();
    while (!empty(panini)) {
        temp = head(panini);
        if (!member(temp, result))
            result = cons(temp, result);
        panini = tail(panini);
    }
    return result;
}
```


Fondamenti di Informatica T-1, 2019/2020 – Modulo 2

Prova d'Esame 5A di Giovedì 9 Luglio 2020 – tempo a disposizione 2h

“main.c”:

```
#include "element.h"
#include "bar.h"

int main() {
    { // Es. 1
        Ingrediente * elenco;
        Ingrediente temp;
        int dim;

        elenco = leggiIng("ingredienti.txt", &dim);
        stampa(elenco, dim);
        temp = trova(elenco, dim, "prosciutto");
        printf("Cercavo prosciutto, ho trovato %s\n", temp.nome);
        temp = trova(elenco, dim, "capperi");
        printf("Cercavo capperi, ho trovato %s\n", temp.nome);
        free(elenco);
    }
    { // Es. 2
        Ingrediente * elenco;
        int dim;
        list panini;

        elenco = leggiIng("ingredienti.txt", &dim);
        ordina(elenco, dim);
        printf("\n\nElenco ingredienti ordinato:\n");
        stampa(elenco, dim);
        panini = leggiPanini("panini.txt", elenco, dim);
        showlist(panini);
        free(elenco);
        freelist(panini);
    }
    { // Es. 3 && 4
        Ingrediente * elenco;
        int dim;
        list panini;
        list paniniSenzaRip;

        elenco = leggiIng("ingredienti.txt", &dim);
        panini = leggiPanini("panini.txt", elenco, dim);
        paniniSenzaRip = eliminaDup(panini);

        showlist(paniniSenzaRip);
        free(elenco);
        freelist(panini);
        freelist(paniniSenzaRip);
    }
    return 0;
}
```

Fondamenti di Informatica T-1, 2019/2020 – Modulo 2

Prova d'Esame 5A di Giovedì 9 Luglio 2020 – tempo a disposizione 2h

“ingredienti.txt”:

prosciutto 80 26.50
funghi_trifolati 20 35.00

mozzarella 50 9.00

scamorza 15 18.00
speck 80 25.99
prosciutto_cotto 80 19.90

“panini.txt”:

Federico prosciutto funghi_trifolati fine
Piero scamorza speck fine
Paola mozzarella prosciutto_cotto fine
Federico speck rucola fine